



# Developing Agility

*Un boletín trimestral para clientes de EAE y AB Suite de Unisys*

## Contenido

- Página 2 **Modernización de EAE con ASP.NET Generator Client Tool**  
Elimine las pantallas verdes siguiendo los cinco pasos de las mejores prácticas que presenta George McGowan para desarrollar interfaces gráficas de usuario (GUI) basadas en el navegador.
- Página 4 **Engineering Corner: las numerosas interfaces de usuario de EAE y AB Suite**  
John Papachristos, de ACUS, analiza métodos avanzados para actualizar los clientes ASP.NET generados.
- Página 9 **Más opciones de GUI: generadores personalizados**  
Una breve descripción de opciones adicionales para mejorar la presentación para el usuario final por medio de generadores personalizados disponibles desde una variedad de fuentes.
- Página 11 **Recursos sobre SOA**  
Cuatro informes técnicos y un estudio de caso ofrecen nuevas perspectivas sobre formas en que las aplicaciones ClearPath pueden participar en una Arquitectura orientada a los servicios (SOA).
- Enlace a **¿Ya leyó ClearPath Connection?**  
[Vea una edición reciente y suscríbese a este boletín electrónico trimestral](#) para mantenerse informado sobre los productos, servicios y eventos de ClearPath. Visite la sección Noticias en eCommunity para obtener más detalles.
- Enlace a **Felicitaciones a los estudiantes de Letonia que han desarrollado destrezas de AB Suite**  
La clase de 2008 incluye 23 personas que estudiaron AB Suite como parte de su plan central de estudios de TI. Vea un artículo sobre este tema en [eCommunity](#).
- Página 11 **Calendario**  
Revise nuestro calendario para obtener información acerca de los próximos eventos.

## Modernización de EAE con ASP.NET Generator Client Tool

Por George E. McGowan, Jr. CPA, presidente y fundador de McGowan Computer Associates

### El costo real de la modernización

La modernización es un tema polémico para muchas organizaciones de TI. Aun cuando la aplicación cumpla todos los requisitos comerciales u organizacionales, las interfaces de usuario con “pantallas verdes” dan la impresión de que el sistema es anticuado y necesita un reemplazo urgente.

Hemos visto a muchas organizaciones asumir el desafío de reemplazar aplicaciones Enterprise Application Environment (EAE) ya ajustadas para sus necesidades por una solución de paquete o varios paquetes. Sin embargo, las posibilidades de implementar algo que proporcione una funcionalidad equivalente son muy escasas. Pocas organizaciones pueden abarcar todas las reglas comerciales contenidas en una aplicación EAE que ha evolucionado con el pasar de los años para satisfacer los cambiantes requisitos del negocio y el mercado.

El precio de este nuevo software esencial para la empresa es alto, generalmente a partir de US\$2 millones. Es muy común que se subestime el costo de la migración. Mapear y trasladar los datos de un sistema a otro es la parte fácil. Lo que eleva el costo es capacitar a los usuarios, calificar el nuevo software, reconstruir las interfaces, realizar pruebas de desempeño exhaustivas con cargas de trabajo de transacciones mixtas e incorporar las funcionalidades “perdidas”. Las organizaciones siempre subestiman el trabajo de probar el nuevo sistema y volver a capacitar a TODOS los usuarios de la aplicación.

Desarrollar una nueva aplicación personalizada no siempre es una opción válida para las organizaciones de TI actuales: los recursos son escasos y las tareas pendientes aumentan cada día.

Considerando todo esto, una aplicación EAE ajustada a sus necesidades específicas es prácticamente invaluable.

### Lo importante es la presentación, no la aplicación

No es necesario dejar a un lado sus aplicaciones EAE si el problema es la presentación. Puede proteger y conservar fácilmente (y seguir afinando y ampliando) las reglas comerciales y al mismo tiempo mejorar la experiencia el usuario final con Client Tools, específicamente ASP.NET Generator. ASP.NET Generator le permite crear una GUI moderna y basada en el navegador, con capacidades que cumplan las expectativas de quienes trabajan con información y dominan el uso de Internet.

Los formularios ASP.NET tienen un funcionamiento excelente, mucho mejor que las Active Server Pages (ASP) originales que llamaban la atención de tantas empresas de EAE hace unos años. ASP.NET presenta las páginas Web universalmente en todos los tipos y niveles de navegadores, así como también en Developer Test. En relación al soporte, no se necesita implementar un software cliente en las estaciones de trabajo. Además, una vez que la organización cambia a ASP.NET, puede elegir mostrar total o parcialmente su aplicación a los usuarios externos a través de Internet.

### Mejores prácticas para desarrollar GUI basadas en el navegador

El uso de ASP.NET Generator es simple pero, como todo desarrollador EAE con experiencia sabe, es esencial contar con el apoyo de las personas interesadas. Hemos utilizado el siguiente método de cinco pasos en muchos proyectos exitosos:

#### 1. Involucrar

Cree un equipo de asesoría que incluya una variedad de usuarios finales, desde aquellos que ingresan datos hasta gerentes y usuarios expertos, y desarrollen en conjunto las pantallas.

#### 2. Experimentar

Los formularios ASP.NET pueden ser una caja de Pandora si no utiliza un método calculado. Trabaje con sus usuarios finales para definir varios tipos de pantallas genéricas, incluyendo un formulario de menú, tabla, consulta e ingreso intensivo de datos. Esto le permite probar las numerosas opciones que ofrece ASP.NET hasta encontrar una funcionalidad adecuada para su entorno. >>

### 3. Simplificar

El hecho de que pueda mostrar los campos como listas desplegables o botones de opción y añadir color y gráfica no significa que tenga que hacerlo. Si su aplicación EAE es un sistema fundamental y serio para la empresa, no ceda ante la tentación de agregar demasiados elementos llamativos. Cuando haya terminado de experimentar, dé un paso atrás para evaluar el valor real que proporcionan las opciones de la nueva GUI para el negocio y simplifíquelas para incluir aquellas que realmente respalden sus necesidades.

### 4. Estandarizar

La mayor parte de los usuarios finales adoptará rápidamente una GUI moderna y fácil de usar. La curva de aprendizaje se minimiza al aplicar una apariencia uniforme, confiable e intuitiva en todos los nuevos formularios ASP.NET. Por eso recomendamos que se tome el tiempo necesario para establecer estándares para las GUI según el resultado de los pasos Experimentar y Simplificar. De tiempo a los usuarios que se resisten al cambio para que trabajen con ellos y cumplan sus necesidades. La transición de un emulador terminal a una aplicación basada en el navegador puede ser difícil para los usuarios actuales. Por otro lado, los nuevos usuarios de la aplicación *esperarán* que sea una aplicación basada en el navegador.

### 5. Modernizar

Con un método organizado como el que se describió anteriormente, tendrá los conocimientos, los estándares y el apoyo de los usuarios finales para actualizar con éxito la presentación de su aplicación EAE utilizando ASP.NET Generator. A pesar de todo, siga los primeros pasos oficiales para que la modernización tenga un impacto mínimo:

- Considere trasladar las transacciones existentes a un navegador Web sin realizar más cambios, manteniendo la navegación y funcionalidad actuales.
- Identifique un subconjunto de pantallas de uso frecuente y aplique selectivamente sus nuevos estándares sólo a esas transacciones, agregando cuadros de lista, botones de opción y otros controles.
- Mantenga estos cambios durante un tiempo, haga los ajustes necesarios y luego aplíquelos a otras transacciones.

El proceso de desarrollo cambiará si decide usar Developer Painter para la implementación de su GUI. El enfoque de las pantallas cambiará de pantallas basadas en caracteres al pintor gráfico. ASP.NET Generator presenta los formularios de una manera muy similar a Developer Painter. Si sólo se accederá a la pantalla a través del navegador Web, no hay motivo para invertir tiempo de desarrollo en hacer que la pantalla basada en caracteres se vea bien. En el pintor de caracteres, ubique los campos en el orden que desea para pasar de uno a otro utilizando las pantallas gráficas. Recuerde que no necesita preocuparse de las etiquetas o encabezados en el pintor basado en caracteres.

#### Ingreso de datos eficiente

¿Quién no recibiría con agrado una GUI moderna basada en el navegador? ¡Todas las personas que ingresan datos en el mundo! Tenga especial cuidado en comprender el proceso que utilizan estas personas para ingresar los datos en las aplicaciones EAE y diseñe una GUI que sea una ayuda y no un obstáculo para ellos. No es necesario usar botones de opción ni listas desplegables con ellos.

### Comenzar con las GUI

Éste es el mejor momento para modernizar su presentación de EAE para el usuario final. ASP.NET Generator es una solución comprobada equivalente a la mayoría de las herramientas de desarrollo ASP.NET que se encuentran en el mercado. Es la mejor manera de actualizar sus pantallas verdes para el siglo XXI en forma rápida y eficiente. Aún no encontramos una necesidad que no se pueda cumplir con las capacidades estándar de ASP.NET Generator. Con esta poderosa herramienta, hemos ayudado a muchas organizaciones a dar pequeños pasos para desarrollar una GUI basada en el navegador, y también pasos gigantes para usar controles sofisticados que traspasan datos a otras aplicaciones que se ejecutan en la estación de trabajo.

Actualícese con un costo de dinero y energía que pueda asumir y conserve todo el valor de su aplicación EAE actual. Modernizar las aplicaciones existentes proporcionará a su organización la tasa de retorno más alta sobre las aplicaciones. Proteja la inversión que ha realizado en sus aplicaciones comenzando a usar ASP.NET Generator.

George E. McGowan, Jr. CPA es presidente y fundador de McGowan Computer Associates, empresa que ha ofrecido servicios de asesoría a organizaciones que utilizan LINC, EAE y Agile Business Suite (AB Suite) durante más de 23 años. Para obtener más información sobre los servicios de asesoría, capacitación, soporte y migración de McGowan Computer Associates, comuníquese con George llamando al 801.446.7100 o enviando un mensaje de correo electrónico a [GMCOWANJR@aol.com](mailto:GMCOWANJR@aol.com).

## Engineering Corner: Las numerosas interfaces de usuario de EAE y AB Suite

*Mejoramiento de los clientes ASP.NET estándar por medio de procesadores y generadores personalizados*

Para continuar con el tema de esta edición sobre las interfaces de usuario, Engineering Corner está dedicado a analizar varios métodos avanzados para actualizar los clientes ASP.NET generados. Le recomendamos que empiece a utilizar Client Tools con sus aplicaciones EAE, inclusive ASP.Net Generator, para mejorar hoy mismo la experiencia del usuario final y ahorrarse este trabajo cuando migre a AB Suite.

Para conocer en detalle las numerosas formas en que puede extender las aplicaciones Web ASP.NET en EAE o AB Suite, hablamos con nuestro experto en la materia, John Papachristos de ACUS. John es ingeniero senior en software y jefe técnico de Client Tools.

**Developing Agility:** ¿Por qué un desarrollador querría hacer más de lo que ya está disponible con ASP.NET Generator estándar?

**John Papachristos (JP):** En nuestro entorno de desarrollo, suministramos un conjunto básico de controles de diseño de formularios en los pintores de pantalla. Dado que se pueden generar múltiples opciones de interfaz de cliente a partir de esta definición de pantalla única, ya sea con EAE o AB Suite, es necesario respetar ciertos límites. No obstante, en ocasiones los desarrolladores desean implementar una interfaz de usuario más sofisticada para el entorno de cliente específico que usarán, como las páginas ASP.NET. Es ahí donde entran en acción las personalizaciones más modernas.

Por ejemplo, con ASP.NET un desarrollador puede agregar una función de visualización de calendario, un control de carga de archivos u otros controles que no ofrecemos automáticamente en nuestros pintores. Siempre nos dicen que nuestro entorno de cliente ASP.NET generado es en realidad un punto de partida para los clientes que quieren sacarle más provecho a su interfaz de navegador. Con Microsoft® Visual Studio®, la aplicación ASP.NET se puede mejorar para que incluya controles adicionales del marco de ASP.NET y se agregan procesos personalizados para satisfacer las necesidades comerciales del cliente.

### ¿Qué es un control personalizado?

Un control es el objeto que se coloca en un formulario (como una página Web) mediante el pintor, para realizar funciones específicas. Los controles comunes incluyen botones, listas desplegables, imágenes, entre otros.

Un control personalizado es aquel que no forma parte del conjunto estándar de controles suministrado por Microsoft. El Procesador de Web Form es un control que ofrece el contenedor para una aplicación de EAE o AB Suite. Maneja las comunicaciones hacia y desde la aplicación y presenta los formularios de transacción (presentación) en el navegador Web.

Entonces, el Procesador de Web Form es una forma de agregar controles u objetos más sofisticados o de realizar procesamientos previos y posteriores a las transacciones entre el cliente y la aplicación sin tener que modificar la aplicación en sí. Además, con el Procesador ya no hay que preocuparse de la sobreescritura, porque siempre se usa el formulario Web Form tal como se generó (salvo en el evento OnAlternateForm, naturalmente). Es una manera excelente de introducir algunos procesos personalizados en el front end o back end, mientras se transfieren datos entre el navegador y el entorno de tiempo de ejecución. >>

**DA:** ¿De qué maneras pueden los desarrolladores mejorar el formulario ASP.NET generado?

**JP:** Hay tres opciones principales. Los desarrolladores pueden:

1. Modificar nuestros archivos de Infraestructura de cliente ASP.NET
2. Usar el Procesador de Web Form
3. Usar el Kit de personalización de generador de cliente, para crear un generador personalizado.

Antes de que analicemos los procesadores y generadores personalizados, veamos cómo los desarrolladores pueden introducir procesos personalizados en varias áreas del marco de cliente ASP.NET mediante nuestros archivos de Infraestructura de cliente.

**DA:** ¿Modificar la infraestructura? Suena difícil... y da un poco de miedo. ¿No lo crees?

**JP:** No es tan difícil como parece. Al realizar cambios en los archivos de infraestructura de nuestro entorno de cliente ASP.NET, los desarrolladores pueden modificar la lógica de presentación y de cliente que se ejecuta en el navegador del equipo del usuario final, al igual que la lógica que se procesa en el servidor Web. Los archivos de infraestructura están incluidos con Client Tools ASP.NET Generator y constituyen la base de nuestro marco de cliente ASP.NET.

Por ejemplo, proporcionamos un archivo de secuencia de comandos llamado CWFRCommonScript.js, que es una recopilación de funciones JavaScript que se ejecutan en el navegador. Dependiendo de las necesidades de una organización determinada, los desarrolladores que dominan JavaScript pueden modificar las funciones o agregar otras a esta secuencia de comandos para invocar procesos personalizados en el navegador. La secuencia de comandos podría actualizarse para captar pulsaciones de teclas y realizar acciones específicas en función de lo que un usuario final escribe, o forzar una tabulación automática al siguiente campo cuando el usuario llegue al final del campo actual. He visto a desarrolladores introducir secuencias de comandos que permiten la interacción entre el formulario del navegador y las aplicaciones externas que se ejecutan en la estación de trabajo local, como el intercambio de datos entre el formulario y Microsoft Word o una aplicación alojada en otra estación de trabajo.

**DA:** Sólo como aclaración, ¿estos cambios se realizan antes de generar el proyecto ASP.NET? Como se está cambiando la “infraestructura”, la próxima vez que se genere, ¿no será necesario modificar nada para volver a aplicar esos cambios?

**JP:** Exactamente. El archivo CEWFRCommonScript.js es parte de nuestro conjunto de archivos de infraestructura y no se sobrescribe con cada generación. Sin embargo, los desarrolladores que optan por modificar los archivos deben tener en cuenta que se proporcionan nuevas versiones en ediciones de Interim Correction (IC). Entonces, si se aplica un nuevo IC se debe tener cuidado de conservar las actualizaciones.

**DA:** Mencionó que había otras formas de modificar los archivos de infraestructura.

**JP:** Otro método para introducir cambios consiste en modificar los archivos de infraestructura basados en el servidor. Uno de estos es IspecView.CS, que define una clase heredada por todos los formularios Web generados para una aplicación. Incluye muchas de las funciones del servidor que se ejecutan cuando se procesa un formulario Web. De esta forma, se presenta la oportunidad de introducir algunos cambios personalizados que influyen en el procesamiento de todos los formularios.

Un desarrollador podría hacer cambios en IspecView.cs para extender los atributos de presentación dinámicos e incluir algunos atributos personalizados. Por ejemplo, se podría efectuar una revisión de rango en campos numéricos o cambiar los colores de determinados botones. Toda la lógica para el procesamiento de atributos dinámicos está contenida en este archivo de infraestructura, que los desarrolladores pueden hacer extensible en función de sus requisitos particulares. >>

**DA:** ¿Dónde se puede encontrar más información acerca de estos cambios en los archivos de infraestructura?

**JP:** Quienes sepan de qué se trata el procesamiento de formularios y C#, podrán deducir fácilmente lo que deben hacer. Además, los archivos de infraestructura incluyen comentarios que ofrecen explicaciones detalladas acerca de las diversas funciones. En el kit de personalización también hallarán información al respecto, pero ese es un artículo que se debe solicitar por separado y no es necesario para realizar cambios en los archivos de infraestructura, que se instalan como parte de Client Tools ASP.NET Generator.

**DA:** ¿Existen otros archivos de infraestructura?

**JP:** Sí, hay muchos otros. Se puede configurar el entorno ASP.NET para que muestre una página de inicio de sesión y los desarrolladores pueden personalizar la página de inicio de sesión que ofrecemos para ajustarla a sus necesidades. También hay otros formularios Web que se pueden personalizar, como las páginas de tiempo de espera y las páginas de error.

Otro punto interesante es el menú contextual del “clic con el botón secundario del mouse”, un control personalizado que se puede actualizar para excluir elementos actuales o introducir procesos personalizados. Sé de una cuenta donde el menú contextual del clic con el botón secundario del mouse se extendió para ofrecer un nuevo elemento de menú, que copia datos de una planilla Microsoft Excel® en los campos Copy.From del formulario Web ASP.NET. El cliente solía usar una función así con su emulador T27 y descubrió que esto no se podía hacer con el entorno de cliente ASP.NET estándar. Al personalizar el menú contextual del clic con el botón secundario del mouse y agregar algunos métodos asociados con JavaScript en CEWFRCommonScript.js, pudieron replicar esta función en sus formularios Web ASP.NET.

**DA:** Entiendo, eso cubre la primera opción para extender los formularios ASP.NET. ¿Qué nos puede decir acerca del uso del Procesador de Web Form?

**JP:** Claro. Otro método para personalizar el entorno ASP.NET consiste en cambiar los archivos generados para ofrecer controles y procesos adicionales. Naturalmente, uno debe ser precavido y asegurarse de administrar los archivos personalizados, ya que las generaciones posteriores podrían sobrescribirlos. Ahí es donde recurrimos al evento “OnAlternateForm” del Procesador de Web Form, porque se puede retener ese evento y visualizar el formulario personalizado para un ISPEC en particular, en lugar del formulario generado “estándar” que se define en la aplicación. Este evento es de utilidad para los clientes cuando tienen ciertos formularios que requiere una personalización más exhaustiva que otros en su aplicación.

**DA:** ¿Qué es un “procesador” y para qué lo usa un desarrollador? ¿Qué clase de mejoras pueden efectuar los desarrolladores en las interfaces de usuario final con los procesadores?

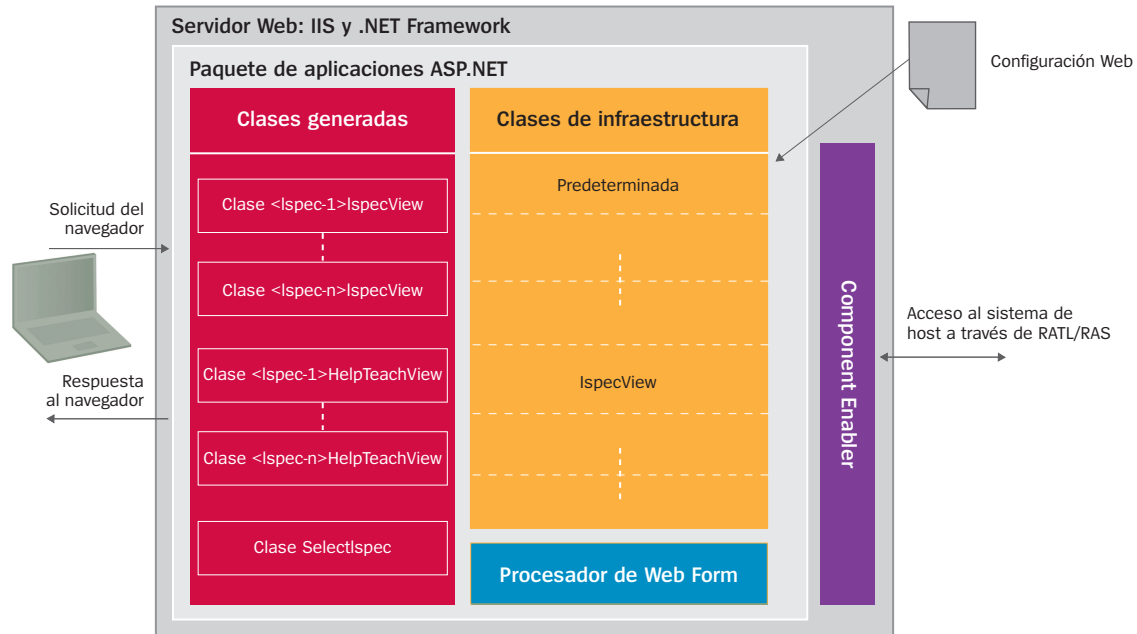
**JP:** En la edición del [primer trimestre de 2006 de este boletín](#), apareció un excelente artículo acerca de los [procesadores](#). Básicamente, el Procesador de Web Form es un control personalizado que permite visualizar los controles de formularios Web que se generan a partir de los formularios ASPEC, es decir, los formularios que se diseñan mediante Developer Painter, y permite que un programador agregue procesos específicos sobre la base de eventos o desencadenantes determinados.

Mencioné el evento OnAlternateForm que le permite reemplazar los formularios que ha personalizado. Existen otros eventos, como OnPreTransaction, OnPostTransaction y OnStatusLine que los desarrolladores pueden utilizar en ciertos puntos al procesar el formulario Web para realizar acciones determinadas. Por ejemplo, son útiles si desea capturar los datos que vienen del sistema de tiempo de ejecución del host antes de que lleguen al navegador y generar valores predeterminados en ciertos campos.

**DA:** ¿Alguna precaución al utilizar el Procesador de Web Form?

**JP:** Lo único que mencionaría es que el procesamiento del tiempo de ejecución puede volverse “costoso” si se implementa mucho procesamiento de formularios a través del Procesador. Recuerde que el Procesador se ejecuta en el servidor Web, no en la estación de trabajo cliente. >>

## Procesamiento del tiempo de ejecución de formularios Web ASP.NET



Esta ilustración muestra la interacción entre las tres capas de una aplicación basada en la Web: navegador Web ejecutándose en una estación de trabajo al formulario Web ASP.NET ejecutándose en el servidor Web y acceso a la aplicación del sistema de host a través de RATL/RAS.

**DA:** Bueno, ya es hora de hablar sobre la tercera opción, usar el Kit de personalización de generador de cliente. ¿Qué es un generador personalizado y de qué manera lo utilizan los desarrolladores?

**JP:** Los desarrolladores usan un generador personalizado cuando tienen muchos cambios de formularios Web, como controles personalizados y procesamientos especiales, y desean que el proceso de aplicar estos cambios se pueda repetir. Si le preocupan los posibles gastos generales de utilizar el Procesador de Web para una personalización extensa, la mejor opción es actualizar directamente los formularios Web generados. Cuando los desarrolladores estén satisfechos con la forma en que funcionan sus formularios Web ASP.NET personalizados, pueden usar nuestro Kit de personalización de generador de cliente para modificar la fuente de ASP.NET Generator y crear su propio generador personalizado. Este generador personalizado producirá formularios Web de la manera que desean los desarrolladores, creará controles adicionales y evitará el procesamiento de tiempo de ejecución que pudiese imponer el Procesador de Web Form.

**DA:** Crear un generador personalizado suena complejo.

**JP:** Esa idea es muy común, pero puedo ayudar a que las personas comiencen a trabajar en sólo unos minutos. Realizamos un taller práctico de 40 minutos en la reunión de usuarios del Foro sobre tecnología de Unisys en Queenstown, Nueva Zelanda. En 10 minutos los asistentes eran capaces de agregar soporte en forma de información sobre herramientas o "tooltips" a los campos utilizando el equipo de personalización. ¡Pruébenlo!

**DA:** ¿Cuándo usaría un desarrollador el Procesador de Web Forms y cuándo crearía o usaría un generador personalizado?

**JP:** Se puede usar el Procesador si no desea realizar cambios en el formulario generado pero sí que se aplique a los formularios que se procesan en tiempo de ejecución. Los desarrolladores deben usar el generador personalizado si quieren hacer cambios en los formularios generados y que ese proceso de cambios se repita. >>

## Las cuatro etapas del proceso de generación



Esta ilustración muestra las cuatro etapas del proceso de generación. La mayor parte del trabajo es realizado en el método `CreatelspecView()`, que ejecuta las distintas clases de "Traverser" que generan las distintas partes de un formulario Web. Una clase Traverser procesa los datos que se llenaron en la memoria por medio de las llamadas realizadas a las interfaces `GenerateProxyLDA` desde EAE o AB Suite Builder. Por lo general, un generador crea una instancia de una o más clases Traverser, dependiendo del resultado que se desee producir para una aplicación cliente en particular.

**DA:** ¿Hay algo que el desarrollador pueda hacer con Client Tools en AB Suite y que no pueda hacer en EAE?

**JP:** No, de hecho tienen la opción de usar la versión AB Suite de Client Tools con EAE. Se pueden utilizar indistintamente entre EAE y AB Suite. Evidentemente, hay capacidades en el pintor de formularios de AB Suite que no existen en EAE. Sin embargo, las capacidades del cliente ASP.NET son las mismas, ya sea que se utilice la configuración predeterminada, el Procesador o el kit de personalización. Es el mismo cliente.

**DA:** Si un desarrollador usa ASP.NET Generator, ¿hay restricciones o limitaciones que debieran tenerse en cuenta, especialmente cuando sus planes son cambiar de EAE a AB Suite?

**JP:** No, no hay restricciones ni limitaciones. Todo lo que puede hacer en EAE migrará sin problemas a AB Suite al cargar el modelo. Los clientes pueden migrar a AB Suite, crear su carpeta de paquete y compilar sus clientes ASP.NET, y la interfaz del usuario en el navegador será idéntica a la de EAE.

**DA:** AB Suite Developer utiliza Visual Studio. ¿Los clientes también deben tener licencia de Visual Studio para emplearlo con Client Tools para las aplicaciones EAE?

**JP:** No, no necesitan licencia de Visual Studio. Sólo se requiere un compilador de C#, que viene con el marco de tiempo de ejecución .NET. Esto significa que puede generar las aplicaciones AB Suite utilizando los archivos de lotes incluidos en AB Suite, y luego implementarlos y emplearlos sin Visual Studio. Todas las demás actividades de personalización que acabo de describir requieren el entorno de desarrollo Visual Studio.

*Muchas gracias a John Papachristos por darnos estos excelentes consejos sobre cómo personalizar los formularios ASP.NET y sobre el procesamiento de formularios.*

## Más opciones de GUI: Generadores personalizados

### Resumen

Este número de *Developing Agility* está completamente dedicado a las numerosas opciones que tienen los desarrolladores para proporcionar una presentación gráfica para el usuario final en las aplicaciones EAE y AB Suite.

En primer lugar, puede usar los pintores de GUI de EAE y AB Suite para definir la presentación para el usuario final como parte del modelo y generarlo para todos los tipos de cliente, inclusive los navegadores Web.

¿Desea hacer más? Puede modificar manualmente el cliente generado, pero no recomendamos este método porque los cambios tendrían que aplicarse de nuevo después de generarlo. Hay muchas opciones más, especialmente si le interesa implementar clientes basados en el navegador. Como se menciona en el artículo de Engineering Corner, puede modificar los archivos de infraestructura del cliente, usar el Procesador de Web Forms para modificar el proyecto ASP.NET que se generó y/o utilizar el Kit de personalización de Client Tools para construir su propio generador personalizado. La mayoría de estas opciones requieren cierto nivel de conocimientos en C# y/o .NET, pero los resultados son impresionantes y se sincronizan perfectamente a medida que su aplicación EAE o AB Suite cambia con el tiempo.

### Espere, ¡hay más!

Varias empresas han desarrollado generadores personalizados y especializados guiados por parámetros. Diseñado para situaciones donde ASP.NET Generator no es suficiente, y cuando la organización no está interesada en crear su propio generador personalizado, estas herramientas facilitan mucho la tarea para los desarrolladores que no tienen grandes conocimientos de Web o .NET para personalizar la GUI. Además, permiten repetir los procesos para que los cambios se vuelvan a aplicar automáticamente en cada generación.

Creado por técnicos expertos que conocen muy bien EAE, AB Suite y las GUI, y que comprenden las necesidades de los usuarios finales, estos generadores personalizados son una opción que bien vale considerar, especialmente si no desea desarrollar conocimientos sobre Web o .NET dentro de la empresa. La siguiente tabla muestra varias de las fuentes más conocidas de generadores personalizados.

Finalmente, la decisión es suya y tiene muchas opciones entre las cuales elegir. ¿Qué es lo adecuado para su organización? Eso depende de dos cosas: qué apariencia desea darle a sus GUI y qué tipo de conocimientos técnicos tiene o desea desarrollar. Tenga en cuenta lo siguiente:

Nombre del generador personalizado Empresa e información de contacto	Descripción
<p><b>ASP.NET Generator para AB Suite y EAE</b>                      Disponible como una contratación de servicio de Unisys Europe al contactar a <a href="mailto:Dominique.Michaut@fr.unisys.com">Dominique.Michaut@fr.unisys.com</a></p>	<p>Gracias a este generador, los desarrolladores pueden mejorar fácilmente las GUI al agregar en forma automática objetos gráficos no compatibles con ASP.NET Generator. Esto se logra al ingresar parámetros en la herramienta del generador personalizado. Esta herramienta puede ser una solución cliente/servidor completa con JavaScript genérico, el que también se puede personalizar.</p> <p>Las posibilidades de control personalizado incluyen:</p> <ul style="list-style-type: none"> <li>• Objeto de calendario</li> <li>• Lista de valores</li> <li>• DataGrid con orden y paginación</li> <li>• Uso de archivo .css (hojas de estilo en cascada)</li> <li>• ¡Y mucho más!</li> </ul> <p>Por su simpleza y eficiencia, Generator ASP.NET for AB Suite and EAE es utilizado por la mayoría de los usuarios de EAE/AB Suite en Francia y también en otros países europeos. &gt;&gt;</p>

Nombre del generador personalizado Empresa e información de contacto	Descripción
<p><b>Interface Builder</b> Disponible en <a href="#">Information Exchange Group (IEG)</a>, Inc.</p>	<p>Interface Builder (IB) es un generador de formularios Web flexible que aprovecha fácilmente las capacidades del navegador y requiere cambios y experiencia mínimos. Desde su lanzamiento en 1999 hasta su versión actual, IBdotNET ha mejorado continuamente y hoy está muy integrado con AB Suite. Los formularios generados por IB se utilizan en aproximadamente 200 organizaciones. Además, tres empresas de desarrollo de software con sede en Norteamérica utilizan el producto.</p> <p>La fortaleza de IB es una infraestructura flexible que permite incorporar funcionalidades gráficas y personalización del cliente o procesamiento de servidor sin necesidad de contar con destrezas avanzadas de programación basada en la Web. La imaginación es el límite.</p> <p>Se puede descargar una copia de IB para evaluación en el sitio Web de IEG. Los términos del período de prueba para evaluación se encuentran en <a href="http://ieg-ib.iegonline.net/ieg-ib.htm">http://ieg-ib.iegonline.net/ieg-ib.htm</a>.</p>
<p><b>Smart Web Client</b> Disponible en <a href="#">McGavin Consulting Group</a></p>	<p>Smart Web Client (SWC) permite a los desarrolladores agregar muchos elementos gráficos modernos a los formularios gráficos de EAE, como botones de calendario, menús desplegables o fichas. La función AJAX Client de SWC permite a los usuarios finales hacer consultas y seleccionar datos de la aplicación sin cambiar Specs. Los usuarios mayores adoran la función de zoom, que expande dinámicamente el tamaño de la visualización de la ventana. SWC también incluye funciones necesarias para ingresar datos rápidamente en operaciones de pantalla de ciclo corto, como las teclas de funciones, los atajos y las teclas numéricas programables.</p> <p>SWC se utiliza en organizaciones de desarrollo de EAE en Nueva Zelanda, Australia y el Reino Unido. Actualmente se está mejorando para que sea compatible con AB Suite, inclusive la actualización automática de SWC de EAE a AB Suite.</p> <p>Visite <a href="http://www.mcgavin.biz">www.mcgavin.biz</a> para obtener una descripción general de SWC o escriba a <a href="mailto:questions@mcgavin.biz">questions@mcgavin.biz</a>.</p>
<p><b>CTC ASP.NET WebForms Generator</b> Disponible en <a href="#">Client Tools Consultancy (CTC)</a></p>	<p>Este generador personalizado apareció destacado en el <a href="#">número del primer trimestre de 2008 de Developing Agility</a>. Al utilizar el Configurador de CTC, los desarrolladores pueden ajustar la apariencia de cada uno de los controles de una pantalla, según se requiera, lo que permite aprovechar totalmente las capacidades de ASP.NET. Esto incluye utilizar controles de terceros para satisfacer las demandas de las interfaces de usuario modernas.</p> <p>Para descargar una versión de prueba del software, visite el <a href="#">sitio Web de CTC</a>.</p>

- Para obtener una GUI excelente que vaya más allá de las antiguas pantallas verdes, la mayoría de las organizaciones de desarrollo están muy satisfechas con Developer Painter y Presentation Client como el mecanismo de visualización.
- Para crear una GUI basada en el navegador con todas las funciones, Client Tools ASP.NET Generator es una opción excelente.
- Las organizaciones de desarrollo que cuentan con conocimientos de C# y ASP.NET, y que tienen la necesidad de crear controles más modernos para sus GUI, pueden considerar realizar mejoras adicionales a los formularios Web a partir de ASP.NET Generator como se describe en la sección Engineering Corner.
- Las organizaciones que desean una interfaz de navegador personalizada, pero no desean desarrollar conocimientos técnicos al interior de la empresa, deben considerar las opciones de generadores personalizados que se describen en la tabla anterior.

Elija la opción que se ajuste mejor a su entorno y necesidades, ¡sus usuarios finales se lo agradecerán!

## Recursos sobre SOA

En un estudio reciente, Galvano Groothandel BV describe cómo crear una SOA aprovechando sus aplicaciones EAE le ha permitido a este mayorista de artículos de fontanería de alta calidad ampliar sus volumen de pedidos de comercio electrónico en un 55%. SOA es un tema polémico y no hay mejor momento que éste para comenzar a explorar de qué manera pueden participar sus aplicaciones EAE y sistemas ClearPath.

SOA le permite obtener un mayor valor de sus aplicaciones comerciales existentes, incluidas aquellas que se basan en AB Suite y EAE, sin poner en riesgo los flujos de ganancias que generan estas aplicaciones.

Unisys desea ayudarlo a aprovechar las oportunidades que presenta la SOA manteniendo el valor comercial inherente a sus aplicaciones estratégicas actuales. Con este fin, publicamos recientemente una serie de informes técnicos sobre la SOA que siguen un progreso lógico desde los conceptos de SOA de alto nivel hasta un análisis del soporte intermedio disponible para que sus aplicaciones ClearPath puedan participar en una SOA.

Los artículos se pueden descargar de la eCommunity:

- [Arquitectura orientada a los servicios: soluciones para la empresa](#)
- [Arquitectura orientada a los servicios: sistemas ClearPath en SOA](#)
- [Productos y estrategias de soporte intermedio para ClearPath OS 2200](#)
- [Productos y estrategias de soporte intermedio para ClearPath MCP](#)

El [estudio de caso de Galvano](#) se puede leer completo en el sitio Web de Unisys.

## Calendario

Próximamente habrá muchas oportunidades para reunirse con los usuarios de AB Suite y EAE y el personal de Unisys en todo el mundo. No olvide revisar [eCommunity](#) para obtener la última información.

Qué	Dónde	Cuándo
Webcast de la actualización de AB Suite, abril de 2008	<a href="#">Disponible a pedido</a>	En cualquier momento
Reuniones de usuarios de Agile Business Suite y EAE (dos fechas a elección)	St. Paul de Vence, Francia	22 al 24 de septiembre de 2008 24 al 26 de septiembre de 2008
Conferencia anual de tecnología <a href="#">UNITE</a>	<a href="#">Caribe Royale Orlando</a> , Orlando, FL	19 al 23 de octubre de 2008

Las especificaciones están sujetas a modificaciones sin previo aviso.

© 2008 Unisys Corporation.

Todos los derechos reservados.

Unisys es una marca registrada de Unisys Corporation. Microsoft, Excel y Visual Studio son marcas registradas de Microsoft Corporation. Todas las otras marcas y productos a los que se hace referencia en este documento se reconocen como marcas registradas o marcas industriales registradas de sus respectivos fabricantes.